4Д

УДК 519.71+681.3

Ю.Б. Деглина, В.С. Денисова, В.А. Козловский
ИПММ НАН Украины, г. Донецк, Украина
kozlovskii@iamm.ac.donetsk.ua

Автоматные алгоритмы синтеза образов

Рассмотрена задача синтеза и анализа простых геометрических объектов – примитивов, представленных в растровом формате. Разработано приложение, реализующее предложенные автоматные алгоритмы для распознавания и синтеза образов.

Необходимость распознавания графических образов возникает при решении широкого спектра технических задач. В целом ряде приложений успех распознавания существенно зависит от того, насколько успешно распознан контур объекта.

В частности, задачи контурного анализа выходят на первый план при переводе инженерно-технической документации из растрового формата в векторный. Системы автоматизированного проектирования (САПР) и инженерного документооборота уже доказали свою состоятельность как эффективный инструмент разработки и поддержки проектной документации в электронной форме. Однако огромное количество инженерно-технических материалов до сих пор хранится в бумажных архивах и обрабатывается устаревшими методами. По некоторым оценкам во всем мире имеется более 8 миллиардов технических изображений, из которых менее 15 % хранится в электронном формате. Несмотря на то, что системы автоматизированного проектирования существуют уже не один десяток лет, более 65 % технических изображений – это бумажные чертежи.

На сегодняшний день существует несколько подходов к решению задачи векторизации:

– *автоматическая* векторизация дает удовлетворительные результаты только на изображениях хорошего качества и, как правило, требует существенной доработки;
– *интерактивная* векторизация, при которой оператор явным образом указывает преобразуемые объекты на растре, дает хорошие результаты, но требует больших временных затрат;
– *гибридные* системы совмещают в себе возможности для работы как с растровыми, так и с векторными изображениями, но также требуют участия оператора в процессе векторизации.

При решении задачи векторизации подзадачи анализа и синтеза примитивов возникают естественным образом.

Целью работы является построение алгоритмов анализа растровых изображений чертежей и алгоритмов синтеза векторного представления примитивов.

Постановка задачи. Исходным описанием будем считать растр – прямоугольное поле размером $m \times n$, разбитое на единичные квадраты, окрашенные белым (0) или черным (1) цветом – на котором представлены два типа примитивов. Ставится задача построения алгоритмов анализа и синтеза примитивов для создания векторного описания растровых изображений и их программная реализация.

Основные определения

Описание объектов будем выполнять в алфавите непроизводных символов $V = \{0,1,2,3,4,5,6,7\}$ цепочечных кодов Фримена, физический смысл которых ясен из рис. 1.

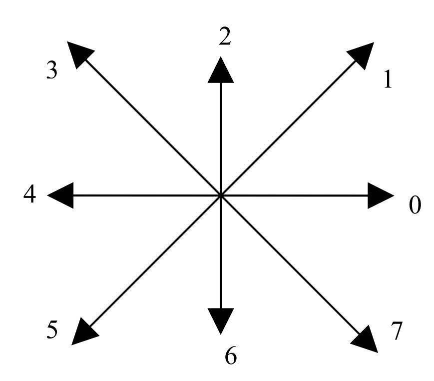


Рисунок 1 – Непроизводные элементы

Поскольку задачи распознавания векторных изображений во многом созвучны задачам распознавания лабиринтов автоматами [1], будем описывать алгоритмы в терминах автоматных моделей. Обозначим множество всех слов конечной длины в произвольном алфавите X, включая пустое слово e, через X^*. Конечным автоматом-преобразователем (автоматом Мили) называется шестерка $A = (S, X, Y, \delta, \lambda, s_0)$, где S – конечное множество состояний, X – конечное множество входных символов (входной алфавит), Y – конечное множество выходных символов (выходной алфавит), δ – функция, определяющая следующее состояние, λ: $S \times X \to Y$ – функция выходов, $s_0 \in S$ – начальное состояние. Функции δ и λ одношаговые и обычным образом распространяются на слова конечной длины. Автоматом-распознавателем назовем пятёрку $A = (S, X, \delta, s_0, F)$, где S, X, δ, s_0 – те же объекты, что и в автомате Мили, а $F \subseteq S$ – множество заключительных состояний. Автомат допускает входное слово p, если $\delta(s_0, p) \in F$. Множество слов, допустимых автоматом A, называется языком, распознаваемым автоматом A. Слово *p* называется периодическим, если его можно представить в виде $p = p_1^k$, где p_1^k обозначает k раз повторенное слово p_1, k = 1, 2, ...; наименьшее такое p_1 называется периодом слова *p*.

Выделение объекта

На этапе выделения объекта растровое изображение сканируется сверху – вниз и слева – направо до нахождения объекта – первой точки, имеющей цвет, отличный от цвета фона (обозначим её (x, y)). Для оконтуривания можно использовать один из вариантов алгоритма жука [2], в качестве которого выступает автомат, блуждающий по клеткам плоскости и имеющий возможность обозревать «ближайших соседей» клетки, в которой он находится. Составление слова описания выполняется на основе непроизводных элементов из алфавита V.

Алгоритм оконтуривания состоит из следующих шагов:

а) поиск «соседей» точки (x, y); «соседями» считаются чёрные клетки с координатами (x–1, y), (x, y–1), (x–1, y–1), (x+1, y), (x, y+1), (x+1, y+1), (x–1, y+1), (x+1, y–1);

б) поиск наиболее подходящего соседа; поиск начинается с точки, которая является следующей (по ходу часовой стрелки) за предыдущей контурной точкой;

в) добавление кода направления, по которому пошел жук, к слову описания образа.

Алгоритм заканчивает свою работу по достижении начальной точки. Результат работы алгоритма – слово описания объекта.

Данный алгоритм может быть реализован в виде конечного автомата, который мы будем называть сканером, блуждающего по среде [1], в качестве которой выступает бинарное поле ограниченного размера $n \times m$. Автомат, находясь в ячейке (i, j), i = 1,…, n, j = 1,…, m, обозревает окрестность этой ячейки, то есть ячейки с координатами (i–1, j–1), (i–1, j), (i, j–1), (i+1, j), (i, j+1), (i+1, j+1), (i–1, j+1), (i+1, j–1). Наблюдая в одной из них элемент изображения (в ячейке нулевое значение цветности), автомат выдаёт на выход соответствующий элемент из алфавита V в зависимости от взаимного расположения наблюдаемой ячейки и ячейки фиксации автомата. Автомат завершает свою работу, если в результате блуждания вдоль контура фигуры он возвращается в стартовую точку.

Алгоритмы автоматного синтеза и распознавания примитивов

Одной из основных особенностей инженерных графических изображений является ограниченность набора используемых примитивов. Фактически, большая часть изображений может быть описана в терминах отрезков прямых (в т.ч. ломаные и многоугольники) и фрагментов эллипсов. В силу того, что анализ и синтез выполняется средствами вычислительной техники, будем рассматривать только отрезки прямых с рациональным угловым коэффициентом. В силу симметрии растровой сетки можно ограничиться рассмотрением только прямых с угловым коэффициентом $k \le 1$.

Для описаний оцифрованных прямых из дискретной геометрии [3] известен результат о том, что для прямых с рациональным угловым коэффициентом $k = \frac{m}{n}$ слово описания периодично. В работе [4] доказана следующая теорема, дающая точное описание оцифрованных прямых в алфавите Фримена. Пусть

$$sign(a) = \begin{cases} 1, & a \ge 0 \\ 0, & a < 0 \end{cases}. \qquad (1)$$

Теорема: Если p(m/n) = $\alpha_1\alpha_2\ldots\alpha_n$, и m ≤ n, то α_i= sign(m-mi mod n), i = 1,…,n.

На основании этих соотношений построена система уравнений автомата, порождающего слово описания прямой с любым угловым коэффициентом $k = \frac{m}{n}$.

$$\begin{cases} s(t) = [s(t-1) + m] \bmod n, \quad s(0) = 0 \\ y(t) = sign(m - s(t-1)), \quad t > 1 \\ y(1) = 1 \end{cases} \qquad (2)$$

Обозначим автомат, описываемый такими уравнениями, через A(m/n). На рис. 2 в качестве примера приведен граф переходов автомата A(3/7), генерирующего слово ‘0101010’, которое описывает прямую y = (3/7)*x.

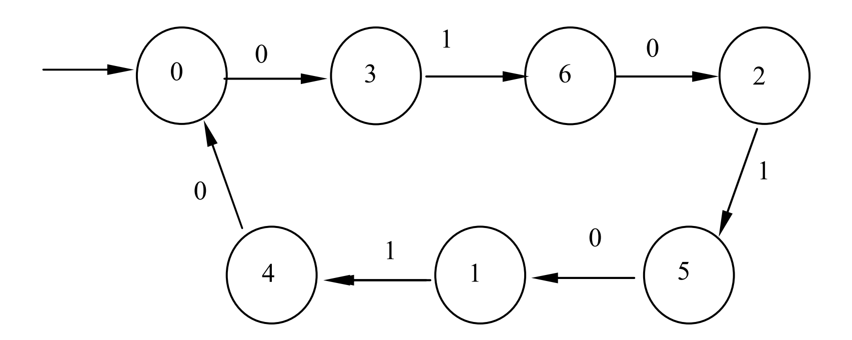


Рисунок 2 – Граф переходов автомата A(3/7)

Отдельно выделяется прямая, соответствующая m = 0. В этом случае прямой, совпадающей с осью Ox, соответствует слово ’00…0’.

В силу симметрии переход к описанию полупрямых y = kx с началом в точке (0, 0) при других углах наклона осуществляется простой перекодировкой.

Таким образом, для любых m и n мы можем синтезировать описание соответствующего отрезка прямой.

Следует особо отметить, что поскольку автомат задается уравнениями, а не таблицами, величины m и n не влияют на сложность реализации и емкостные характеристики алгоритма.

Работа над построением автоматов, синтезирующих окружности и эллипсы еще ведется, и представляется вероятным, что для генерации линий второго порядка могут потребоваться более сложные связки автоматов.

Все прямые с k = m/n, 0 ≤ m ≤ n, описываются словами p(m/n), и можно построить генерирующий все эти последовательности автомат A(n), входным алфавитом которого можно считать множество X={0, 1,…, n,_}. Каждое из этих значений соответствует некоторому значению m, с помощью которого автомат настраивается на генерацию слова p(m/n).

Состояния автомата A(m/n) переобозначим и получим множество {(m, 0), (m, 1),…,(m, n–1)}. Переходы осуществляются, как и в исходном автомате A(m/n), по любому $x \in X$. Множество состояний тогда есть $S = \bigcup_{m=0}^{n} S_{\frac{m}{n}} \bigcup \{s_0\}$ и $\delta(s_0, m) = (m, 0)$.

Обозначим через r_m начальное состояние автомата A(m/n). Тогда автомат A(n) можно представить в виде следующей схемы, приведенной на рис. 3.

На основе автомата A(n) можно построить автомат-распознаватель, который будет сравнивать предложенное слово p(m/n) последовательно со всеми словами, генерируемыми автоматом A(n). Так как множество слов p(m/n) конечно и они периодические, то объединение таких слов является регулярным языком и существует автомат, распознающий этот язык.

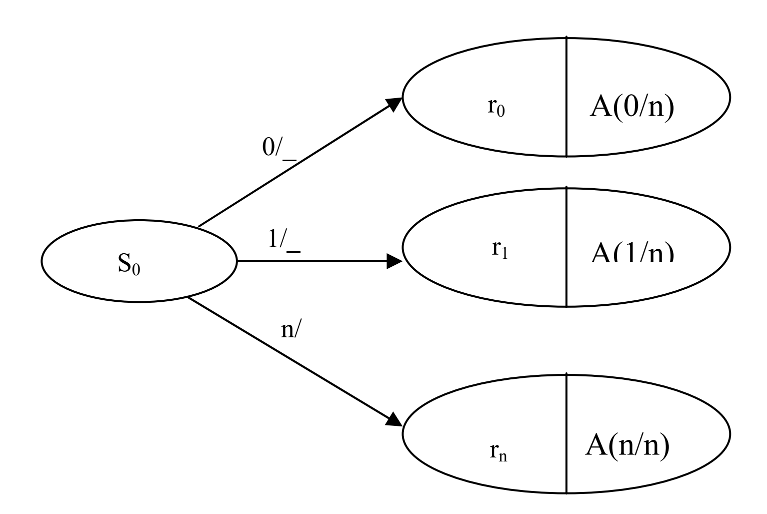


Рисунок 3 – Схема автомата A(n)

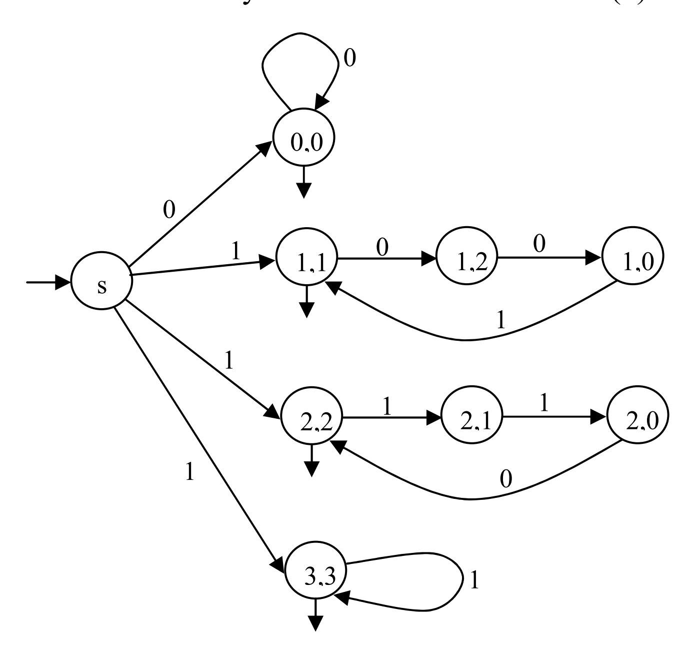


Рисунок 4 – Граф переходов автомата R(3)

Автомат-распознаватель R(n), распознающий множество всех периодических слов $\left(p\left(\frac{m}{n}\right)\right)^k$, k > 0, и их начальных отрезков, проще представить в виде недетерминированного автомата, полученного определённым обращением по выходам автомата A(n). Из состояния s_0 переходы осуществляются по всем дугам по входу, равному 1 в состояние (m, m), а из состояния (m, i) в состояние (m,j) осуществляется переход по входу α, если в A(n) был переход из (m, i) в (m,j) с выходом α. По всем неопределённым остальным входам из всех состояний осуществляется переход в особое «мертвое» состояние s, которое переходит в себя по всем входам. Заключительными состояниями объявляются все состояния автомата, кроме начального s_0 и «мертвого» s. Такой автомат распознаёт слова, описывающие все отрезки полупрямых, начинающихся в начале координат (подразумевается, что один из концов отрезка тоже лежит в начале координат). Ниже приведен пример автомата R(n) при n = 3.

Программная реализация алгоритмов

На основании описанных алгоритмов была реализована версия программной системы распознавания растровых изображений. В системе реализован описанный выше алгоритм оконтуривания объектов с генерацией представлений в цепочечных кодах Фримена, а также алгоритм распознавания примитивов типа «отрезок прямой». Система предполагает возможность расширения и встраивания дополнительных алгоритмов.

Для распознавания используется два варианта алгоритмов. Первым из них является способ проверки полного соответствия описания, сгенерированного автоматом-сканером, описанию прямой, сгенерированному автоматом, порождающим описания идеальных прямых. Второй способ является способом «ускоренного» распознавания, при котором слово описания проверяется на соответствие некоторым известным характеристикам дискретных прямых и совпадение описания с идеальным прообразом проверяется только на некотором заранее выбранном количестве точек. Второй способ работает несколько быстрее, но не может гарантировать правильность результатов в целом ряде случаев.

Заключение

В работе предложены автоматные алгоритмы синтеза образов и распознавания отдельных примитивов. На этой основе разработано приложение, реализующее распознавание отдельных типов примитивов как предложенным автоматным способом, так и ускоренным способом на основе некоторых известных об оцифрованных прямых фактов, который, однако, не дает стопроцентной точности распознавания.

Предполагается, что методы, изложенные в статье, могут быть развиты для построения комплексной системы, которая будет решать задачи анализа изображений, формирования образа и последующей генерации изображений в векторном формате, причем предусматривается как построение собственного языка синтеза изображений, так и возможность экспорта в некоторые существующие форматы векторной инженерной графики.

Литература

1. Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов – М.: Наука. Гл. ред. физ.-мат. лит., 1985. – 320 с.
2. Абламейко С.В., Лагуновский Д.М. Обработка изображений: технология, методы, применение. – Минск: Амалфея, 2000. – 304 с.
3. Rosenfeld A., Klette R. Digital Straightness // Electronic Notes in Theoretical Computer Science. – 2001. – № 46. – 32 с. – Режим доступа: http://www.elsevier.nl/locate/entcs/volume46.html.
4. Деглина Ю.Б., Козловский В.А., Костогрыз К.А. Автоматное распознавание оцифрованных многоугольников // Искусственный интеллект. – 2004. – № 3. – С. 443-452.

Ю.Б. Дегліна, В.С. Денісова, В.А. Козловський

Автоматні алгоритми синтезу образів

Розглянуто задачу синтезу та аналізу простих геометричних об'єктів – примітивів, що представлені у растровому форматі. Розроблено систему, що реалізує запропоновані автоматні алгоритми розпізнавання та синтезу образів.

Yu.B. Deglina, V.S. Denisova, V.A. Kozlovskii

Automata Algorithms of Pattern Synthesis

The problem of analysis and synthesis of simple geometric objects called primitives presented in raster format is considered. An application implementing the proposed automata algorithms is developed.

Статья поступила в редакцию 20.07.2008.